

Projet Année 3 Big Data/IA/**Web**

Partie Développement Web

Projet encadré par :

- Ayoub KARINE <ayoub.karine@isen-ouest.yncrea.fr>
- Antoine DESRIER <antoine.desrier@isen-ouest.yncrea.fr>

Contexte du projet

Objectif

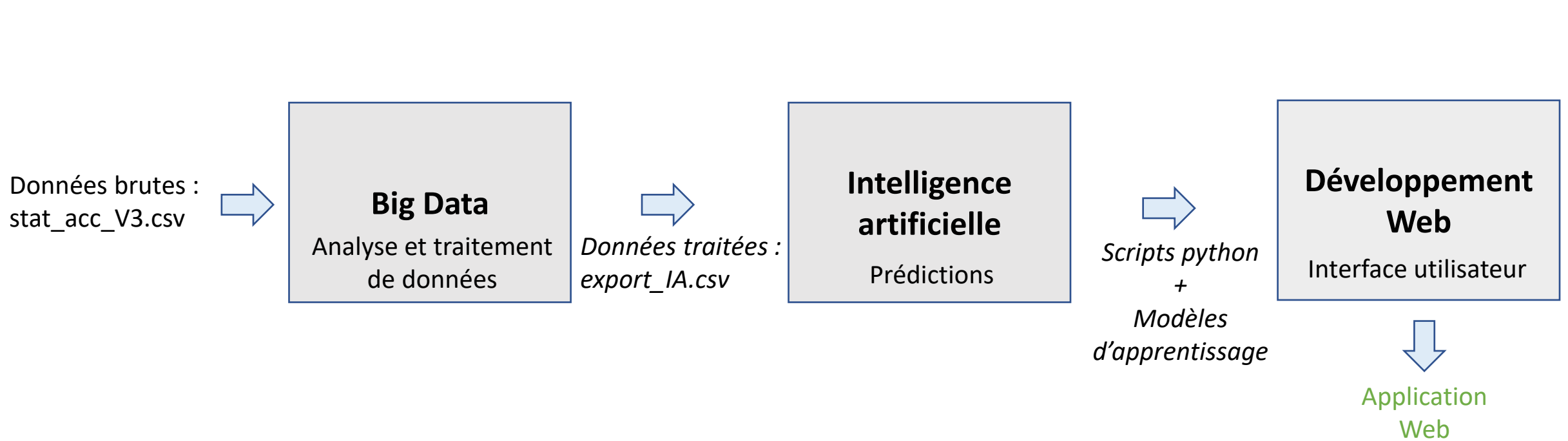
Concevoir et développer une application d'étude des accidents de la route

Approfondir les compétences acquises dans les modules *Big Data*, *Intelligence Artificielle*, *Développement Web et Base de Données* à travers une application complète de traitements et de visualisation de données concernant les accidents corporels de la circulation routière en France.

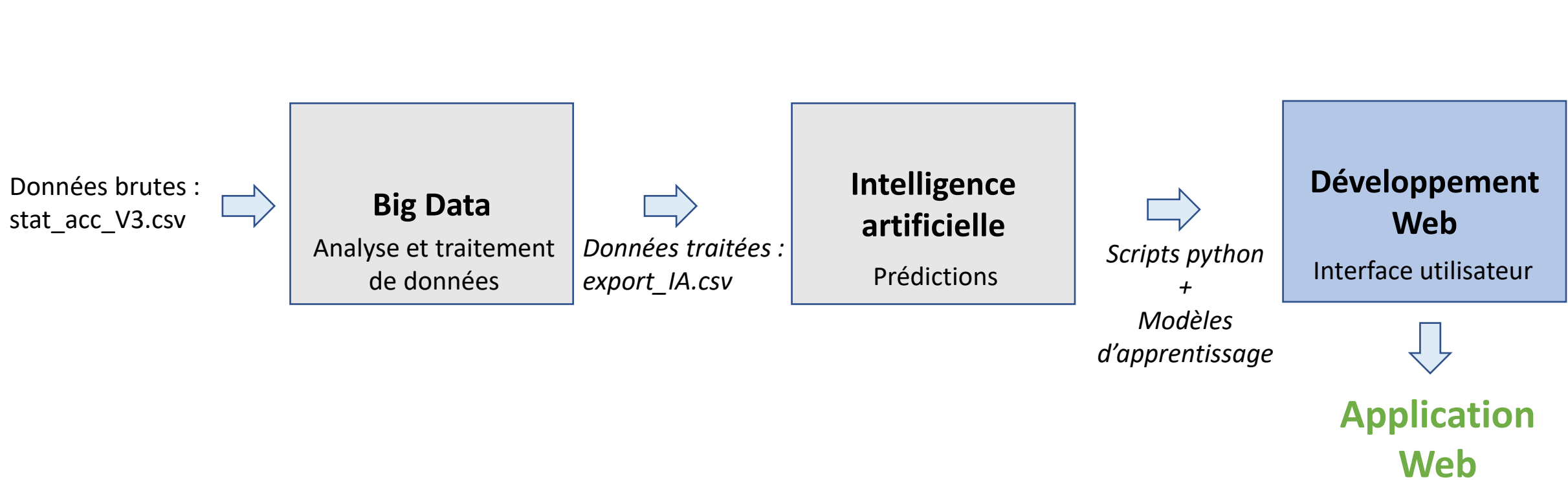
Objectifs de la partie Développement Web :

- Programmation web coté client (*front-end*) :
 - Créer une maquette visuelle d'un site web
 - Programmer les éléments de la maquette visuelle en *HTML*
 - Programmer le style de la maquette visuelle en *CSS*
 - Modifier le comportement de la page web en *JavaScript*
 - Manipuler *AJAX*
- Programmation web coté serveur (*back-end*) :
 - Créer un code *PHP* qui encapsule les requêtes permettant d'interagir avec la base de données
 - Traiter les réponses des requêtes en *PHP* et envoyer des réponses au client

Déroulement du projet



Déroulement du projet



Processus de développement

Analyse :

- Maquette du site web (*FIGMA*)
- Modèle Conceptuel de Donnée (*MCD*)
- Requêtes client-serveur associées aux pages présentées

Conception et développement :

- *HTML*
- *CSS*
- *JavaScript*
- *PHP*
- *SQL*

Test

Recette fonctionnelle

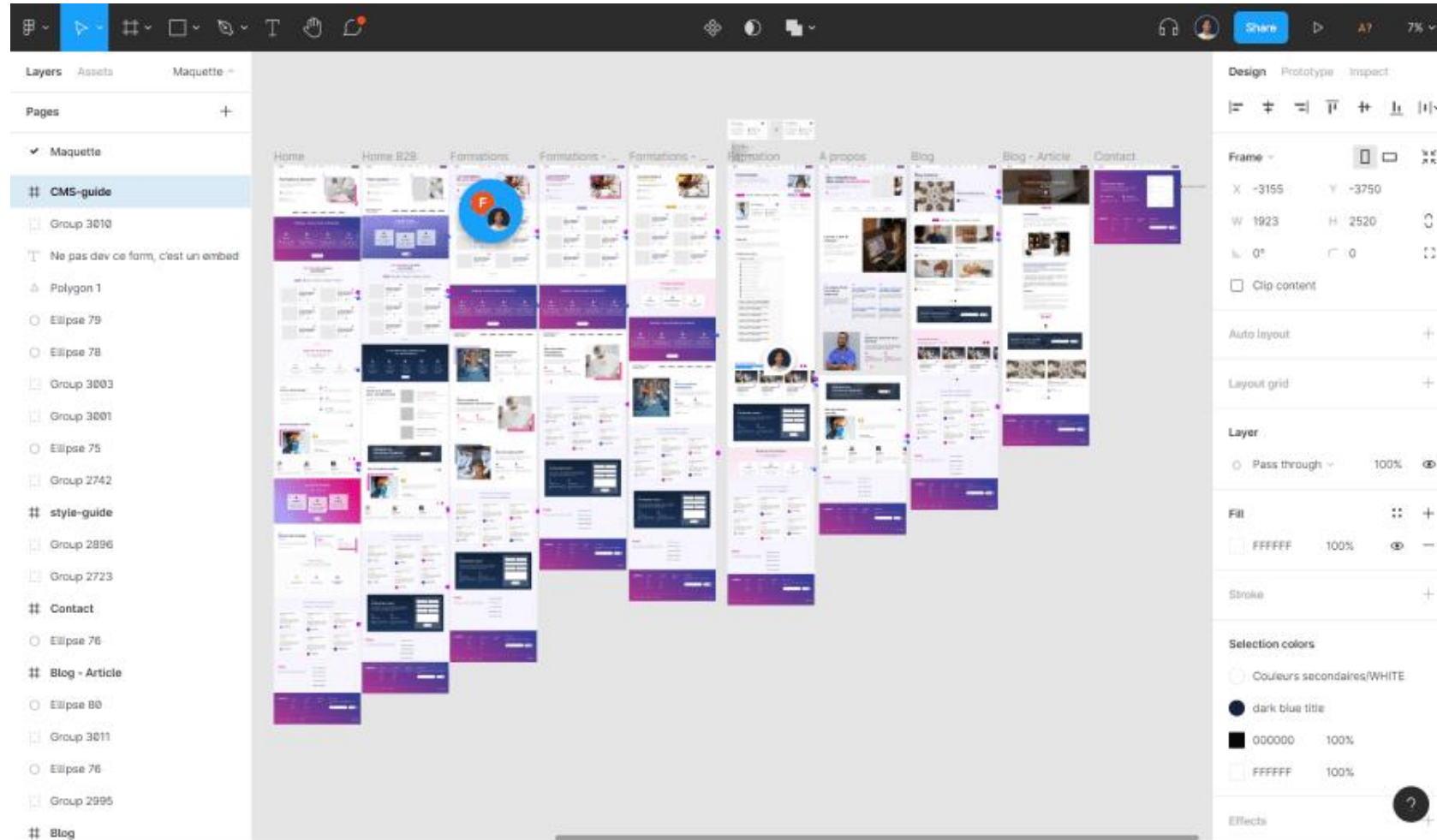
Cahier des charges

Maquette de l'application web

Figma

Préparation du squelette de votre application web :

Réalisation de la maquette en ligne avec *Figma* :



Conception de la base de données de l'application web :

Les nouveaux accidents ajoutés à l'aide de l'application web devront être stockés dans une base de données :

- Créez les entités nécessaires
- Ajouter les champs nécessaires
- Réfléchissez aux types et à la taille de chaque champ
- Créez les relations adéquates
- Réfléchissez aux cardinalités de vos relations
- Définissez vos clés primaires

→ **Attendus : le modèle conceptuel de données (MCD) de votre application web en *PDF***

Attention

C'est à vous de définir les entités et les relations de votre modèle conceptuel de donnée en étudiant avec attention le cahier de charges fourni dans ce document.

5 fonctionnalités principales sont attendues :

1. Page d'accueil
 2. Ajout d'accidents
 3. Visualisation des accidents ajoutés dans un tableau et sur une carte
 4. Prédiction du *cluster* des accidents entrés (à partir du script *Python* de l'IA)
 5. Prédiction de la gravité d'un l'accident (à partir des scripts *Python* de l'IA)
- + Import des données initiales avec un script Python (**pour les CIR uniquement ou en bonus**)
 - + Options de filtrage des accidents pour la visualisation (**pour les CIR uniquement ou en bonus**)
 - + Authentification pour l'ajout d'un accident (**en bonus**)

Attention

Les accidents des fichiers CSV (utilisées en Big Data et IA) ne seront pas insérés dans votre base de données web sauf si le bonus 1 est réalisé.

Fonctionnalité 1

Page d'accueil

Ajout d'une page d'accueil :

- Ajout d'un menu pour naviguer entre les différentes pages du site
- Descriptif rapide du projet
- Image représentant le projet

→ **Attendu : une page d'accueil qui s'affiche quand l'utilisateur entre l'url du site web**



Ajouter un nouvel accident dans la base de données :

Un formulaire doit permettre d'ajouter un accident avec au minimum les informations suivantes :

- L'âge du conducteur
- La date et l'heure
- La ville
- La latitude et la longitude
- Les conditions atmosphériques (en menu)
- La luminosité de la scène (en menu)
- L'état de la route (en menu)
- L'état d'utilisation de la ceinture de sécurité (en menu)

Utilisez le même encodage
adopté en Big Data

→ **Attendu : une page web contenant un formulaire permettant d'ajouter un nouvel accident**

Pour ces nouveaux accidents, la gravité ne doit pas être fournie.

Fonctionnalité 3

Visualisation des accidents ajoutés sur une carte

Visualisation des accidents de la base de données web :

Tous les accidents de la base de données web devront apparaître :

- Dans un tableau avec au minimum:
 - L'âge du conducteur
 - La date et l'heure
 - La ville
 - La latitude et la longitude
 - Les conditions atmosphériques
 - La luminosité de la scène
 - L'état de la route
 - L'état d'utilisation de la ceinture de sécurité
- Sur une carte :
 - La position de chaque accident devra être visible sur la carte
 - Les détails d'un accident seront visibles lors du survol du point avec la souris

→ **Attendu : une page web contenant un tableau et une carte avec les accidents de la base de données**

Conseil

Il est fortement conseillé d'utiliser la bibliothèque *plotly* : <https://plotly.com/javascript/> pour l'affichage des données sur une carte.

Fonctionnalité 4

Prédiction du *cluster* des accidents

Prédiction du *cluster* des accidents entrés dans la base de données :

Les actions à suivre pour prédire les clusters sont :

1. Clic sur un bouton « Prédire les clusters » en bas de la page web de visualisation des accidents
2. Ouverture d'une nouvelle page
3. Appel coté serveur de votre script *Python* permettant de prédire les *clusters*
4. Affichage des accidents sur une carte
5. Coloriage du *cluster* d'appartenance

→ **Attendu : une page web affichant sur une carte les accidents avec leur *cluster* d'appartenance**



Fonctionnalité 4

Prédiction du *cluster* des accidents

Prédiction du *cluster* des accidents entrés dans la base de données web :

Les actions à suivre pour prédire les clusters sont :

1. Clic sur un bouton « Prédire les clusters » en bas de la page web de visualisation des accidents
2. Ouverture d'une nouvelle page
3. Appel coté serveur de votre script *Python* permettant de prédire les *clusters*
4. Affichage des accidents sur une carte
5. Coloriage du *cluster* d'appartenance

→ **Attendu : une page web affichant sur une carte les accidents avec leur *cluster* d'appartenance**



Pour appeler votre script *Python* à partir de votre script PHP, vous pouvez utiliser la fonction `system` :

<https://www.php.net/manual/fr/function.exec.php>

```
import sys

def somme(a, b):
    return float(a) + float(b)

s = somme(sys.argv[1], sys.argv[2])
print(s)
```



```
<?php
· exec("python script.py 5 3", $output);
· print_r($output);
?>
```

Array ([0] => 8.0)

Fonctionnalité 5

Prédiction de la gravité d'un l'accident

Prédiction de la gravité d'un accident entré dans la base de données web :

Les actions à suivre pour prédire la gravité d'un accident en utilisant la méthode *KNN* et les méthodes de « haut niveau » sont :

1. Sélection d'un accident dans le tableau à l'aide d'un bouton radio
 2. Clic sur un bouton « Prédire la gravité » en bas de la page web de visualisation des accidents
 3. Ouverture d'une nouvelle page
 4. Appel coté serveur de vos scripts *Python* permettant de prédire la gravité
 5. Affichage de la gravité de l'accident avec un comparatif des méthodes : *KNN*, *SVM*, Random Forest, *MLP*
- **Attendu : une page web affichant la gravité d'un accident avec les différentes méthodes de classification**

Attention

Plusieurs scripts *Python* seront à appeler en fonction de la méthode à utiliser. Méthode simple : *KNN* et méthodes de « haut niveau » : *SVM*, Random Forest, *MLP*

Cahier des charges

Technologies à utiliser

Partie *front-end* :

- Exclusivement avec les technologies *HTML*, *JavaScript* et *CSS*



- Les bibliothèques *Plotly* et *Mapbox*, et uniquement celles-ci, peuvent être utilisées



Partie *back-end* :

- Exclusivement en *PHP*



- SGBD* de type *MySQL*



- Communications avec le *font-end* avec *AJAX*

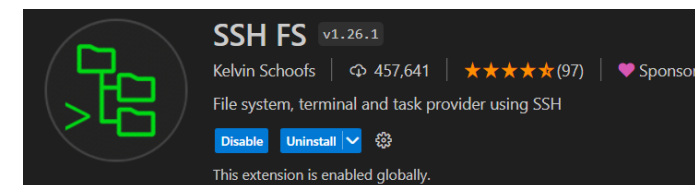
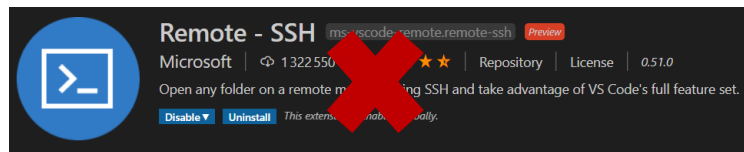


- Échange de données avec le *font-end* en *JSON*



Cahier des charges

Déploiement



[Lien utile](#)

Cahier des charges

Technologies à utiliser

Mise en page :

Il faut que la mise en page du *front* prenne en compte un en-tête et un pied de page commun à toutes les pages.

Accès :

L'accès au site web se fera via un serveur web hébergé sur `etuXXX.isen-ouest.fr`

L'application web doit être :

- Ergonomique : facilité d'utilisation, homogénéité des informations, respect des normes utilisées pour le Web
- Évolutive : possibilité de rajouter de nouvelles fonctionnalités

Le code doit être :

- Correctement architecturé
- Réexploitable : code lisible, code bien commenté, noms de variables/fonctions explicites
- Séparé en plusieurs fichiers (par fonctionnalités)

Livrables et évaluations

Travail en trinôme :

- Chaque étudiant dans le trinôme connaît l'ensemble du projet : **utiliser le gestionnaire de code *Git***
- Attention à bien se répartir le travail en prévoyant les tâches de chacun avec un **diagramme de Gantt**

Ressources externes :

- Tous les documents sont autorisés
- Attention à utiliser avec une grande précaution tout document extérieur : site de vulgarisation, forum, code d'autrui

Documentation du projet :

- Au fur et à mesure
- Standardisée
- Livraison de code ou de documents :
 - Ne pas attendre la dernière minute pour poster un livrable
 - Préparer des livrables intermédiaires (surtout pour les sources)
 - Sauvegarder régulièrement vos données

Format de l'archive :

Archive *ZIP*, *TGZ*, *7ZIP*, pas de *RAR* : projetweb_groupeX.zip (remplacer X par votre numéro de trinôme)

Le rendu final doit contenir :

- L'intégralité de vos codes sources commenté avec vos ressources (images, bases de données...) **ainsi que les données de l'IA** (scripts *Python*, modèles...) nécessaire au fonctionnement de votre application web
- Présentation au format PDF :
 - gestion de projet (Gantt + qui a fait quoi)
 - attendus
 - lien d'accès au site web
- Lien Figma au format PDF
- Votre *MCD* au format *PDF*
- Vidéo de 2 min montrant le fonctionnement du site web

Remarques :

- Malus possible sur l'un des membres du groupe si l'investissement est jugé trop faible
- Possibilité d'être interrogé durant le projet de façon individuelle
- Plagiat sévèrement sanctionné pour TOUS les membres du/des groupe(s)

QCM le 29/06/2023 à partir de 17h00 :

- QCM de 20 minutes à points négatifs
- Évaluation individuelle

Soutenance le 30/06/2023 à partir de 08h00 :

- Soutenance de 20 minutes = Présentation + Démo + Questions
- Présentation en trinôme (pensez à vous répartir la parole)
- Présentez l'essentiel de votre projet

Code :

- Rendu de l'intégralité de vos codes sources avec les ressources associées
- Rendu de votre maquette Figma
- Rendu de votre *MCD*
- Vidéo de 2 min montrant le fonctionnement du site web

Barème indicatif :

- Présentation 40% -- QCM individuel 20% -- Code/maquette/MCD 40%